

# 컴파일러 과제4

## 보고서

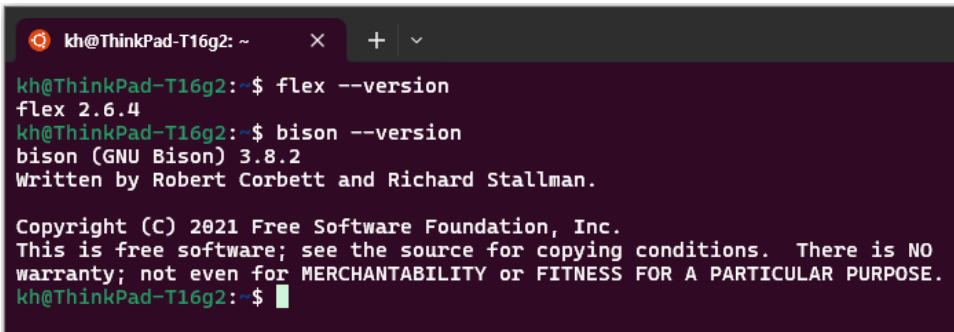
20181755 이건희

2024년 10월



## 1. 개발 환경

기존과 동일한 wsl2 상의 Ubuntu 22 LTS, gcc 11, gdb 12이다.



```
kh@ThinkPad-T16g2: ~  
kh@ThinkPad-T16g2:~$ flex --version  
flex 2.6.4  
kh@ThinkPad-T16g2:~$ bison --version  
bison (GNU Bison) 3.8.2  
Written by Robert Corbett and Richard Stallman.  
  
Copyright (C) 2021 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
kh@ThinkPad-T16g2:~$
```

lex로 flex, yacc로 bison을 사용하였다.

부록의 코드를 전부 따라 치는 것이 쉬운 일이 아니라서, 아이폰의 ocr 기능으로 스캔하고, 오타자를 책과 비교 해가며 찾아 교정하였다.

(프로젝트의 규모가 커지다보니 소스코드 전부를 부록으로 싣기에 너무 길어져서, 본 과제부터는 부득이하게 소스코드만 따로 모아놓은 압축파일을 같이 첨부하겠습니다.)

## 2. 파일 및 수정사항

### 2-1. type.h

lms에 올라온 파일을 그대로 사용하였다.

### 2-2. print.c, print.h (부록 5)

lms에 올라온 원본은 한 파일 내에 prototype과 그 정의 두 부분으로 이루어져 있었는데, 이것을 .c, .h로 분리했다(아래에서 자세히 설명함). 또한 switch의 default 분기에서 error를 출력할 때 디버깅을 위해 어떤 함수인지 출력 하도록 했다.

### 2-3. common\_header.h

컴파일 시 헤더가 중복되어 포함되면 문제가 발생한다. 만약 헤더를 분리하지 않는다 하면

main.c : [stdio.h](#), [stdlib.h](#), print.h, syntax\_analysis.h

print.c : [stdio.h](#), [stdlib.h](#), [type.h](#)

syntax\_analysis.c : [stdio.h](#), [stdlib.h](#), [type.h](#), string.h, y.tab.h

위와 같이 충돌이 발생한다.

```
Debug syntax_parser
print.h common_header.h x syntax_analysis.h ty
1  #ifndef __CSTDLIB_STDIO__
2  #define __CSTDLIB_STDIO__
3  #include <stdio.h>
4  #endif
5
6  #ifndef __CSTDLIB_STDLIB__
7  #define __CSTDLIB_STDLIB__
8  #include <stdlib.h>
9  #endif
10
11 #ifndef __TYPE_HEADER__
12 #define __TYPE_HEADER__
13 #include "type.h"
14 #endif
```

따라서

1. 새로운 헤더파일 common\_header.h를 ifndef~define~endif를 사용하여 중복되지 않게 stdio.h, stdlib.h, type.h를 포함하도록 제작
2. print.h, syntax\_analysis.h를 만들어서 정의된 함수의 선언을 따로 분리
3. main.c, print.h, syntax\_analysis.h에서 common\_header.h 포함  
(main.c에서 type.h가 필요가 없는 것은 알고 있으나 편의를 위해...)

위와 같이 한다면 print.c, syntax\_analysis.c는 각각 동명의 헤더 파일만 포함하기만 하면 되며, print.h, syntax\_analysis.h는 헤더 중복에 대한 우려 없이 단순히 common\_header.h를 포함하기만 하면 되어, 파일 간 관계가 간결해진다.

+추가) 아키텍처 비트 수 문제 수정

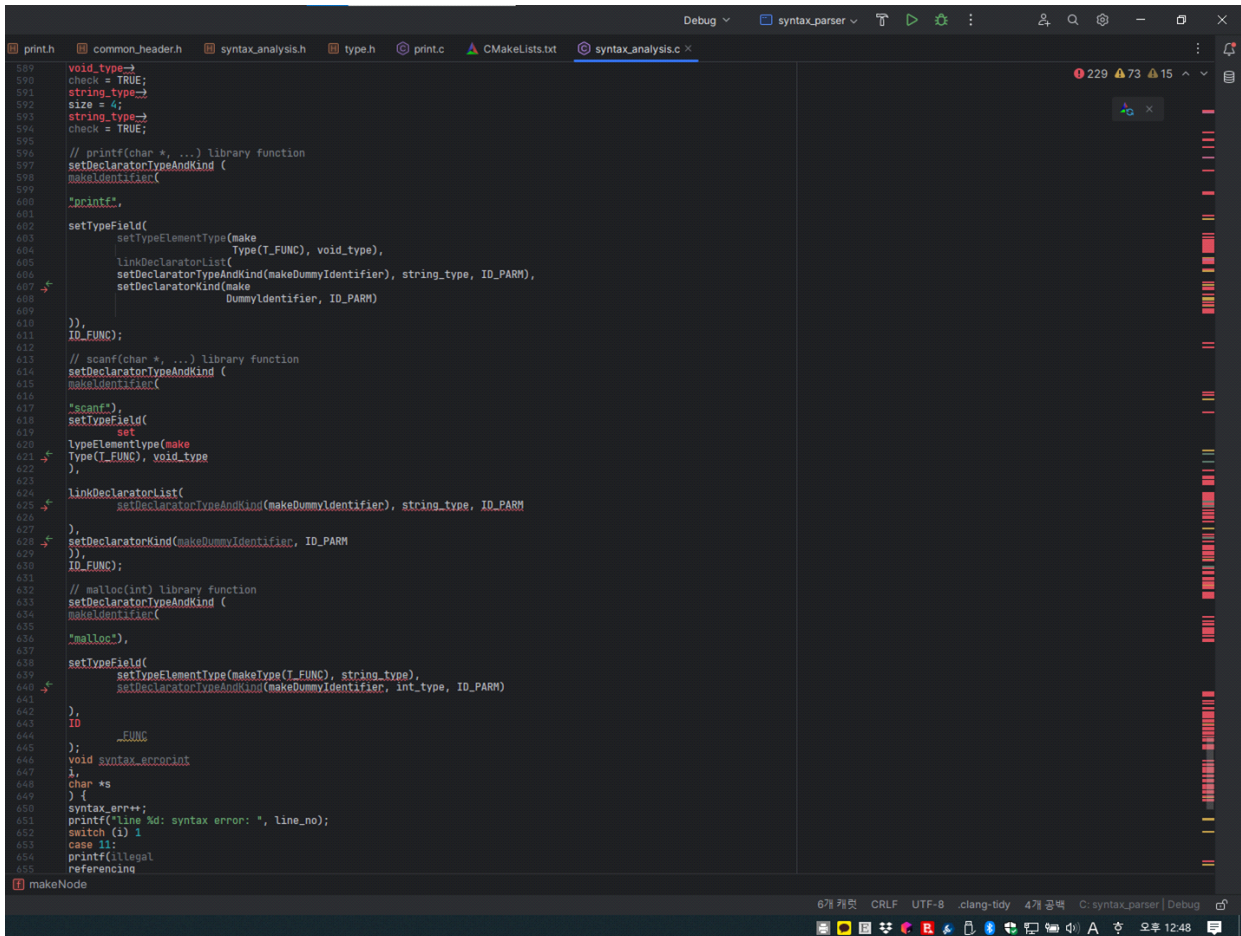
```
14 #endif
15
16 #define YYSTYPE_IS_DECLARED 1
17 typedef long YYSTYPE;
```

11/6 수업 중 아키텍처 비트 수가 다른 문제를 교수님께서 말씀하신 것처럼 하여, common\_header.h에 추가하였다.

```
kh@ThinkPad-T16g2:~/ctest/241107$ uname -mpio
x86_64 x86_64 x86_64 GNU/Linux
kh@ThinkPad-T16g2:~/ctest/241107$
```

실제로 사용하고 있는 wsl2 우분투 가상머신은 64비트이다.

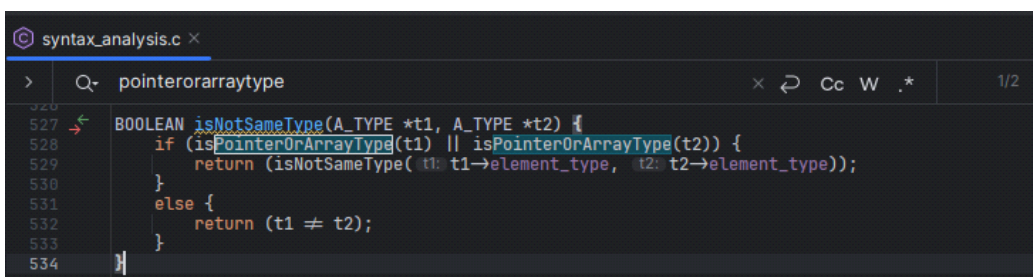
## 2-4. syntax\_analysis.c, syntax\_analysis.h (부록 4)



```
589 void_type=>
590 check = TRUE;
591 string_type=>
592 size = 4;
593 string_type=>
594 check = TRUE;
595
596 // printf(char *, ...) library function
597 setDeclaratorTypeAndKind (
598 makeIdentifier(
599 "printf",
600
601 setTypeField(
602 setTypeElementType(make
603 Type(T_FUNC), void_type),
604 linkDeclaratorList(
605 setDeclaratorTypeAndKind(makeDummyIdentifier), string_type, ID_PARM),
606 setDeclaratorKind(make
607 DummyIdentifier, ID_PARM)
608
609 )),
610 ID_FUNC);
611
612 // scanf(char *, ...) library function
613 setDeclaratorTypeAndKind (
614 makeIdentifier(
615 "scanf",
616
617 setTypeField(
618 set
619 typeElementType(make
620 Type(T_FUNC), void_type
621 ),
622
623 linkDeclaratorList(
624 setDeclaratorTypeAndKind(makeDummyIdentifier), string_type, ID_PARM
625 ),
626 setDeclaratorKind(makeDummyIdentifier, ID_PARM
627 )),
628 ID_FUNC);
629
630 // malloc(int) library function
631 setDeclaratorTypeAndKind (
632 makeIdentifier(
633 "malloc",
634
635 setTypeField(
636 setTypeElementType(makeType(T_FUNC), string_type),
637 setDeclaratorTypeAndKind(makeDummyIdentifier, int_type, ID_PARM)
638 ),
639 ID
640 _FUNC
641 );
642 void syntax_error() {
643 char *s
644 ) {
645 syntax_err++;
646 printf("line %d: syntax error: ", line_no);
647 switch (i) {
648 case 1:
649 printf("illegal
650 referencing
651 makeNode
```

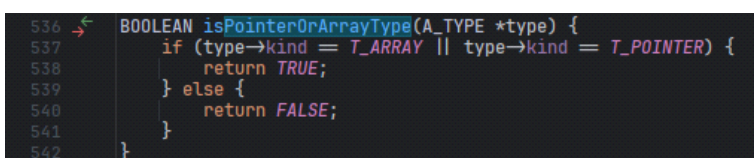
상기하였듯 아이폰의 ocr기능으로 부록 4를 스캔하여 오타자를 잡은 뒤에 수정을 했다. 또한 원본은 함수의 prototype이 선언되어있고 그 이후로 함수의 정의가 나오는데, 이를 .c와 .h로 분리하였다. 분리를 하지 않는다면 yacc.y의 규칙에서 사용되는 함수들의 프로토타입을 전부 yacc.y의 초반에 선언해야 하는데 그러면 코드가 많이 지저분해진다. 이외 수정 사항은 아래와 같다.

### 2-4-1. isPointerOrArrayType()



```
527 BOOLEAN isNotSameType(A_TYPE *t1, A_TYPE *t2) {
528     if (isPointerOrArrayType(t1) || isPointerOrArrayType(t2)) {
529         return (isNotSameType(t1->element_type, t2->element_type));
530     }
531     else {
532         return (t1 != t2);
533     }
534 }
```

isPointerOrArrayType()은 isNotSameType()에서 두 번 사용된다. 부록4의 앞부분에는 prototype으로 선언되어 있으나, 함수 자체는 정의되어있지 않았다. 해당 함수에 대한 설명은 7장 (시멘틱 분석기) 148쪽에서 찾을 수 있었다. 설명에 따라서 완성한 함수는 다음과 같다.



```
536 BOOLEAN isPointerOrArrayType(A_TYPE *type) {
537     if (type->kind == T_ARRAY || type->kind == T_POINTER) {
538         return TRUE;
539     } else {
540         return FALSE;
541     }
542 }
```

## 2-4-2. setDefaultSpecifier()

“A\_TYPE \*t”은 함수 내에서 미사용되기에 삭제

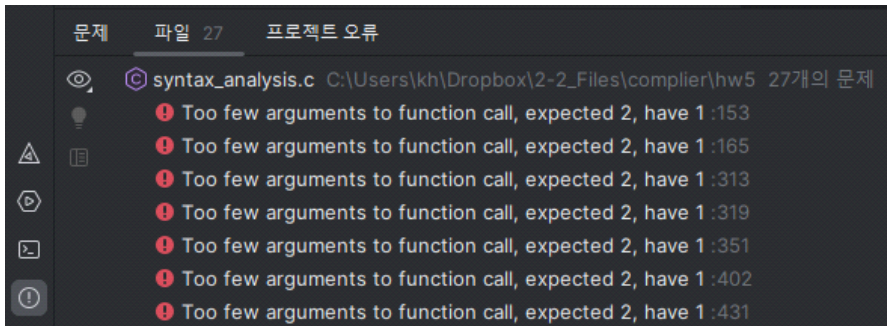
## 2-4-3. syntax\_error()

- prototype과 실제 정의의 매개변수

부록4의 앞부분에서 prototype은 매개변수가 없이 선언되어 있으나, 정의는 매개변수가 있다.

```
72  
73 void syntax_error(int i, char *s);  
74  
75
```

그래서 선언을 수정하니



당연하게도 케이스 14, 20, 21, 23, 24, 25에서 에러가 발생하였고

```
160  
161 if (s) {  
162     if (p→stor) {  
163         if (s = p→stor) { ;  
164     }  
165     else {  
166         syntax_error( i: 24, s: NULL);  
167     }  
168 }  
169 else {  
170     p→stor = s;  
171 }  
172  
173 return p;  
174 }
```

이들을 전부 위와 같이 두 번째 인자로 NULL을 전달하게끔 수정했다.

- 오류 발생 시 프로그램 종료

파싱 중 에러가 감지되면 올바른 트리의 구성이 불가능하기에 더 이상의 트리 생성을 멈추고 곧바로 프로그램을 종료시켜야 한다. 그러나 부록4의 syntax\_error()에서는 프로그램을 종료하는 코드가 없이 단순히 syntax\_err를 후위증가하기만 하여, main()에서 전역변수 syntax\_err를 검사하여 0이 syntax\_err가 0이 아닐 경우 종료하는 식이었다. 따라서 syntax\_error()의 가장 마지막에 exit(1)을 추가했다.

## 2-5. yacc.y

과제 4에서 제출 한 파일에서 다음과 같은 수정을 하였다.

### 과제5 print\_node() segfault

```
38      | declaration
39      ;
40
41      function_definition
42      : declaration_specifiers declarator compound_statement
43      | declarator compound_statement
44      ;
45
46      declaration_list_opt
47      : // no option
48      | declaration_list
49      ;
50
51      declaration_list
52      : declaration
53      | declaration_list declaration
54      ;
55
```

디버깅해 본 결과 정작 문제였던 것은  
사진처럼 과제4에서 사용했던 yacc.y를 그대로 사용해서, 함수와 관련된 부분이 전혀 없어서였습니다  
공유되는 전역변수 A\_NODE\* root가 NULL인 이유입니다

제가 시험공부를 하는 중간에 급하게 과제를 하면서 실수를 했습니다

빌드 시에 오류가 나서 교수님께 이렇게 메일을 드린 적이 있었고, yacc.y를 많이 수정했다.

### - syntax\_analysis.h 포함

11/6 수업에서 교수님이 yacc.y의 초반에서 아래의 규칙에 사용되는 함수들의 prototype을 미리 선언해두어야 한다 하셨다. 그러나 바로 위에서 말했듯 해당 함수들은 syntax\_analysis.c, .h로 분리를 해두었고, 따라서 yacc.y에서 필요한 함수들은 단순히 syntax\_analysis.h를 포함하는 것만으로 사용이 가능하다.

### - common\_header.h 포함

위에서 헤더의 중복 포함을 방지와 아키텍처 비트 수 차이를 해결하기 위해 common\_header.h를 만들었다. y.tab.c는 stdio.h, stdlib.h, type.h를 필요로 한다. 이들은 전부 common\_header.h에 포함된 내용이므로 해당 헤더 파일을 포함하도록 한다.

### - main() 함수 삭제

최종 프로그램에서는 main()을 별개의 파일에서 실행하고 싶었다. 직전의 과제 4에서 최종적으로 나오는 프로그램의 main()은 yacc.y --[yacc]--> y.tab.c에서 나온다. 여기에 덧붙혀 어떻게든 완성하려면 할 수도 있겠으나 직관적이지 못하므로, prototype과 정의를 삭제하였다. main()은 main.c에서 나온다.

### - yyerror() 수정

과제 4의 보고서에서도 yyerror()를 lms상 공지를 따라서 만들었기 때문에 개선해야 하는 한계점으로 적었었다. 부록 3을 참고하여 yyerror()를 수정하였다.

## 2-6. lex.l

과제 4에서 제출 한 파일에서 다음과 같은 수정을 하였다.

- common\_header.h를 포함  
yacc.y와 같은 이유로 포함하였다.

- makeString() 추가  
부록 2를 그대로 사용하였다.

- checkIdentifier() 수정

과제 4에서 checkIdentifier()는 교재 본문의 내용대로 간단하게 만들었다. 새로운 checkIdentifier()는 부록 2를 토대로 약간만 바꿔 사용하였는데, char \*t는 함수 내에서 사용되지 않아서 삭제했다.

## 2-7. main.c (부록 9)

부록 9를 사용하되, semantic analysis에 관련된 부분은 주석처리 하였다. 이 부분은 추후 과제에서 사용될 것이다. 프로그램은 아직 syntax analysis만 진행하므로, main()은

initialize() - symantic\_analysis.h에서

yyparse() - y.tab.h에서

print\_ast() - print.h에서

를 하고 종료된다.

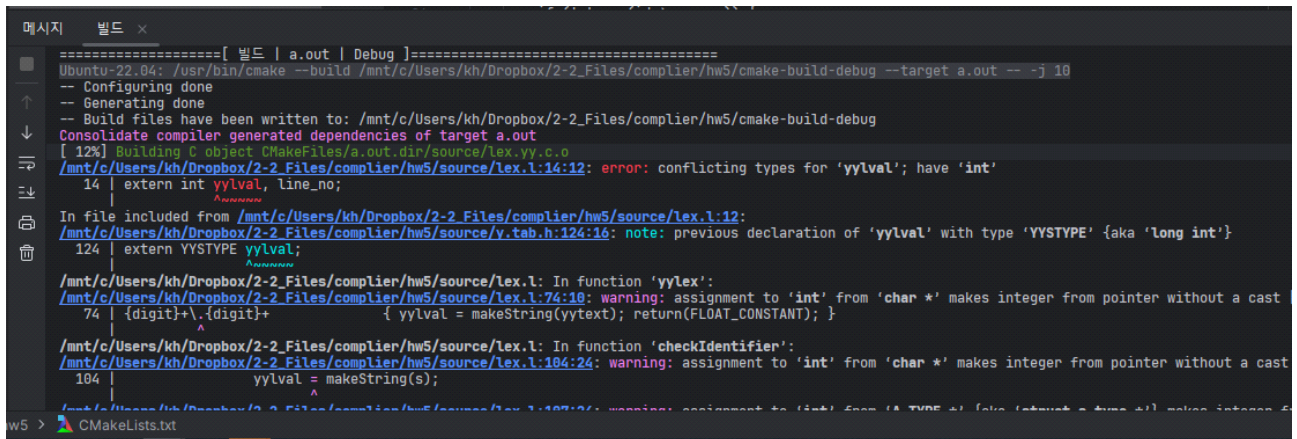
semantic analysis와 관련된 부분은 모두 주석처리 하였다(아마 마지막 과제에서는 전부 사용할 것이라 생각된다).



### 3. 빌드 & 디버그

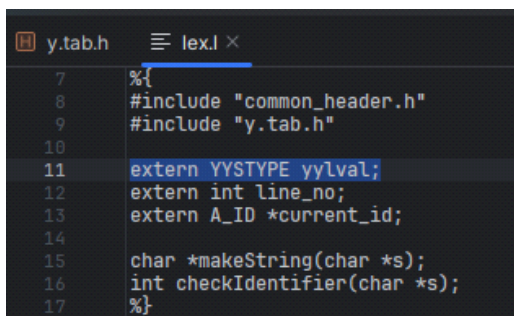
터미널 상 작업을 계속 하는것이 번거로워서 ide에서 CMake를 사용하여 빌드하였다.

#### 3-1. build (1)



```
===== [ 빌드 | a.out | Debug ] =====
Ubuntu:22.04: /usr/bin/cmake --build /mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/cmake-build-debug --target a.out -- -j 10
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/cmake-build-debug
Consolidate compiler generated dependencies of target a.out
[ 12%] Building C object CMakeFiles/a.out.dir/source/lex.yy.c.o
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l:14:12: error: conflicting types for 'yyval'; have 'int'
14 | extern int yyval, line_no;
   |           ^~~~~~
In file included from /mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l:12:
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/y.tab.h:124:16: note: previous declaration of 'yyval' with type 'YYSTYPE' {aka 'long int'}
124 | extern YYSTYPE yyval;
     |           ^~~~~~
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l: In function 'yylex':
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l:74:18: warning: assignment to 'int' from 'char *' makes integer from pointer without a cast
74 | {digit}+\. {digit}+
   | {digit}+\. {digit}+
   | ^~~~~~
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l: In function 'checkIdentifier':
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l:184:24: warning: assignment to 'int' from 'char *' makes integer from pointer without a cast
184 | yyval = makeString(s);
     | ^~~~~~
/mnt/c/Users/kh/Dropbox/2-2_Files/complier/hw5/source/lex.l:184:24: warning: assignment to 'int' from 'char *' makes integer from pointer without a cast
184 | yyval = makeString(s);
     | ^~~~~~
hw5 > CMakeLists.txt
```

빌드 중 lex.yy.c.o를 만드는데 오류가 발생했다. yacc.y를 yacc을 돌려 나온 y.tab.h에서 yyval의 자료형은 YYSTYPE인데, YYSTYPE은 common\_header.h에서 long형으로 정의하였다. 그러나 lex.l에서 외부변수 yyval을 가져오기를 int로 가져와서 에러가 난 것이다.



```
y.tab.h
lex.l
7 | %{
8 | #include "common_header.h"
9 | #include "y.tab.h"
10 |
11 | extern YYSTYPE yyval;
12 | extern int line_no;
13 | extern A_ID *current_id;
14 |
15 | char *makeString(char *s);
16 | int checkIdentifier(char *s);
17 | %}
```

따라서 lex.l에서 외부변수 yyval의 자료형을 int에서 YYSTYPE로 다시 지정해주었다.

### 3-2. build (2)

```
[ 75%] Linking C executable a.out
/usr/bin/ld: CMakeFiles/a.out.dir/source/y.tab.c.o: in function `yparse':
/mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:109: undefined reference to `setTypeStructOrEnumIdentifier'
/usr/bin/ld: /mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:115: undefined reference to `getTypeOfStructOrEnumRefIdentifier'
/usr/bin/ld: /mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:129: undefined reference to `setStructDeclaratorListSpecifier'
/usr/bin/ld: /mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:142: undefined reference to `setTypeStructOrEnumIdentifier'
/usr/bin/ld: /mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:146: undefined reference to `getTypeOfStructOrEnumRefIdentifier'
collect2: error: ld returned 1 exit status
gmake[3]: *** [CMakeFiles/a.out.dir/build.make:173: a.out] Error 1
gmake[2]: *** [CMakeFiles/Makefile2:83: CMakeFiles/a.out.dir/all] Error 2
gmake[1]: *** [CMakeFiles/Makefile2:90: CMakeFiles/a.out.dir/rule] Error 2
gmake: *** [Makefile:124: a.out] Error 2
```

최종적으로 a.out을 만드는데

getTypeOfStructOrEnumRefIdentifier()

setStructDeclaratorListSpecifier()

setTypeStructOrEnumIdentifier()

의 세 함수가 없다 오류가 발생했다. syntax\_analysis.c와 헤더파일을 다시 봐야 할 것이다.

```
main.c y.tab.c syntax_analysis.h x print.h lex.yy.c yacc.y type
28
29 A_ID *getIdentifiedDeclared(char *);
30 A_TYPE *getTypeOfStructOrEnumRefIdentifier(T_KIND, char *, ID_KIND);
31
32 A_ID *setDeclaratorInit(A_ID *, A_NODE *);
33
34 A_ID *setDeclaratorKind(A_ID *, ID_KIND);
35
36 A_ID *setDeclaratorType(A_ID *, A_TYPE *);
37
38 A_ID *setDeclaratorElementType(A_ID *, A_TYPE *);
39
40 A_ID *setDeclaratorTypeAndKind(A_ID *, A_TYPE *, ID_KIND);
41
42 A_ID *setDeclaratorListSpecifier(A_ID *, A_SPECIFIER *);
43
44 A_ID *setFunctionDeclaratorSpecifier(A_ID *, A_SPECIFIER *);
45
46 A_ID *setFunctionDeclaratorBody(A_ID *, A_NODE *);
47
48 A_ID *setParameterDeclaratorSpecifier(A_ID *, A_SPECIFIER *);
49
50 A_ID *setStructDeclaratorListSpecifier(A_ID *, A_TYPE *);
51
52 A_TYPE *setTypeSpecifier(A_TYPE *, A_SPECIFIER *);
53
54 A_TYPE *setTypeElementType(A_TYPE *, A_TYPE *);
55
56 A_TYPE *setTypeField(A_TYPE *, A_ID *);
57
58 A_TYPE *setTypeExpr(A_TYPE *, A_NODE *);
59
60 A_TYPE *setTypeAndKindOfDeclarator(A_TYPE *, ID_KIND, A_ID *);
61
62 A_TYPE *setTypeStructOrEnumIdentifier(T_KIND, char *, ID_KIND);
63
```

헤더파일을 보니 위의 세 함수가 정의되어있지 않다 나왔다.

```
233 // get type of struct identifier
234 A_TYPE *getTypeOfStructOrEnumRefIdentifier(T_KIND k, char *s, ID_KIND kk) {
235     A_TYPE *t;
236     A_ID *id;
237
```

getTypeOfStructOrEnumRefIdentifier()는 I가 소문자였다.

```

411
412 A_ID *setStructDeclaratorListSpecifier(A_ID *id, A_TYPE *t) {
413     A_ID *a;

```

setStructDeclaratorListSpecifier()는  
setStructDeclaratorListSpecifier로 오타가 두 개 있었다.

```

479
480 // set type of struct identifier
481 A_TYPE *setTypeStructOrEnumIdentifier(T_KIND k, char *s, ID_KIND kk) {
482     A_TYPE *t;
483     A_ID *id, *a;

```

setTypeStructOrEnumIdentifier()는 I가 소문자였다.

```

main.c y.tab.c syntax_analysis.h x syntax_analysis.c print.h lex.yy.c yacc.y type.h common_header.h
4
5 A_NODE *makeNode(NODE_NAME, A_NODE *, A_NODE *, A_NODE *);
6
7 A_NODE *makeNodeList(NODE_NAME, A_NODE *, A_NODE *);
8
9 A_ID *makeIdentifier(char *);
10
11 A_ID *makeDummyIdentifier();
12
13 A_TYPE *makeType(T_KIND);
14
15 A_SPECIFIER *makeSpecifier(A_TYPE *, S_KIND);
16
17 A_ID *searchIdentifier(char *, A_ID *);
18
19 A_ID *searchIdentifierAtCurrentLevel(char *, A_ID *);
20
21 A_SPECIFIER *updateSpecifier(A_SPECIFIER *, A_TYPE *, S_KIND);
22
23 void checkForwardReference();
24
25 void setDefaultSpecifier(A_SPECIFIER *);
26
27 A_ID *linkDeclaratorList(A_ID *, A_ID *);
28
29 A_ID *getIdentifierDeclared(char *);
30
31 A_TYPE *getTypeOfStructOrEnumRefIdentifier(T_KIND, char *, ID_KIND);
32
33 A_ID *setDeclaratorInit(A_ID *, A_NODE *);
34
35 A_ID *setDeclaratorKind(A_ID *, ID_KIND);
36
37 A_ID *setDeclaratorType(A_ID *, A_TYPE *);
38
39 A_ID *setDeclaratorElementType(A_ID *, A_TYPE *);
40
41 A_ID *setDeclaratorTypeAndKind(A_ID *, A_TYPE *, ID_KIND);
42
43 A_ID *setDeclaratorListSpecifier(A_ID *, A_SPECIFIER *);
44
45 A_ID *setFunctionDeclaratorSpecifier(A_ID *, A_SPECIFIER *);
46
47 A_ID *setFunctionDeclaratorBody(A_ID *, A_NODE *);
48
49 A_ID *setParameterDeclaratorSpecifier(A_ID *, A_SPECIFIER *);
50
51 A_ID *setStructDeclaratorListSpecifier(A_ID *, A_TYPE *);
52
53 A_TYPE *setTypeSpecifier(A_TYPE *, A_SPECIFIER *);
54
55 A_TYPE *setTypeElementType(A_TYPE *, A_TYPE *);
56
57 A_TYPE *setTypeField(A_TYPE *, A_ID *);
58
59 A_TYPE *setTypeExpr(A_TYPE *, A_NODE *);
60
61 A_TYPE *setTypeAndKindOfDeclarator(A_TYPE *, ID_KIND, A_ID *);
62
63 A_TYPE *setTypeStructOrEnumIdentifier(T_KIND, char *, ID_KIND);
64
65 BOOLEAN isNotSameFormalParameters(A_ID *, A_ID *);
66
67 BOOLEAN isNotSameType(A_TYPE *, A_TYPE *);
68
69 BOOLEAN isPointerOrArrayType(A_TYPE *);
70
isPointerOrArrayType

```

모든 함수가 정의되었음을 확인하고, 다시 빌드를 했다.

### 3-3. build (3)

```

메시지 빌드 x
In file included from /mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:3:
/mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/syntax_analysis.h:53:30: note: expected 'A_TYPE *' {aka 'struct s_type *'} but argument is of type 'YYSTYPE {aka long int}'
53 | A_TYPE *setTypeSpecifier(A_TYPE *, A_SPECIFIER *);
   | ~~~~~^
/mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:404:92: warning: passing argument 2 of 'setTypeSpecifier' makes pointer from integer without a cast
404 | : declaration_specifiers abstract_declarator_opt {$$ = setTypeSpecifier($2, $1);}
   | ~~~~~^
In file included from /mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:3:
/mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/syntax_analysis.h:53:40: note: expected 'A_SPECIFIER *' but argument is of type 'YYSTYPE {aka long int}'
53 | A_TYPE *setTypeSpecifier(A_TYPE *, A_SPECIFIER *);
   | ~~~~~^
/mnt/c/Users/kh/Dropbox/2-2 Files/complier/hw5/source/yacc.y:404:61: warning: assignment to 'YYSTYPE' {aka 'long int'} from 'A_TYPE *' makes pointer from integer without a cast
404 | : declaration_specifiers abstract_declarator_opt {$$ = setTypeSpecifier($2, $1);}
   | ~~~~~^
[ 75%] Linking C executable a.out
[100%] Built target a.out
빌드 이(가) 완료되었습니다.
hw5 > CMakeLists.txt

```

프로그램이 빌드는 되었으나 경고가 많이 발생했는데, 대부분이 print.c에서 발생한 것이라 무시해도 무방할 듯 하다.

```

BODY
N_STMT_COMPOUND (0,0)
(ID="arr") TYPE:e1786aa0 KIND:VAR SPEC=AU
TYPE
ARRAY
INDEX
N_EXP_INT_CONST (0,0)
5
(none)
ELEMENT_TYPE
(int)
TNTT

```

최종적으로 테스트를 하려 실행시켜보니 배열에서 출력될 이유가 없는 “(none)”이 출력되었다.

```

360     prt_expression( node, t->expr, s);
361 }
362 else {
363     print_space( s);
364 }
365 printf( format: "(none)\n");
366 print_space(s);
367 printf( format: "| ELEMENT_TYPE\n");
368 prt_A_TYPE( t->element_type, s);
369 break;

```

```

if (t->expr)
    prt_expression(t->expr,s+2);
else
    print_space(s+2); printf("(none)\n");
print_space(s); printf("| ELEMENT_TYPE\n");
prt_A_TYPE(t->element_type,s+2);

```

올려주신 print.c를 가져오면서 실수를 했다. 이를 else 안으로 포함시켰다.



```

| | | | | (ID= size ) TYPE:930da210 KIND:PARAM SPEC=NULL LEV=1 VAL=0 ADDR=
| | | | | TYPE
| | | | | (int)
| | | | | TYPE
| | | | | (void)| (ID="bubblesort") TYPE:93dc0d00 KIND:FUNC SPEC=NULL LE
| | | | | TYPE
| | | | | FUNCTION
| | | | | PARAMETER

```

또한 null이나 void일 경우 출력형식이 이상했다.

```

        printf("(char %d)\n",t->size)
    else if (t==void_type)
        printf("(void)");
    else if (t->kind==T_NULL)
        printf("(null)");
    else if (t->prt)

```

이는 원본 파일도 틀리게 되어있었다.

```

314
315 → void prt_A_TYPE(A_TYPE *t, int s) {
316     print_space(s);
317     if (t == int_type) {
318         printf( format: "(int)\n");
319     }
320     else if (t == float_type) {
321         printf( format: "(float)\n");
322     }
323     else if (t == char_type) {
324         printf( format: "(char %d)\n", t->size);
325     }
326     else if (t == void_type) {
327         printf( format: "(void)");
328     }
329     else if (t->kind == T_NULL) {
330         printf( format: "(null)");
331     }

```

print.c에서 출력 뒤에 개행 문자를 덧붙혔다.

```

CL hw5 - print.c
CMakeLists.txt yacc.y print.c x type.h
> Q- printf 38/38
382     prt_A_ID_LIST( id: t->field, (s) s + 2);
383     printf( format: "FUNCTION\n");
384     print_space(s);
385     printf( format: "| PARAMETER\n");
386     prt_A_ID_LIST( id: t->field, (s) s + 2);
387     print_space(s);
388     printf( format: "| TYPE\n");
389     prt_A_TYPE( t: t->element_type, (s) s + 2);

```

print.c에서 printf()를 전수조사하여 이전의 경우 이외의 개행 문자가 없는 경우를 찾았으나 없었다.

### 3-4. build (4)

The screenshot shows an IDE with a project named 'hw5'. The left sidebar displays a file tree with 'testdir' containing files '1.c' through '10.c'. The main editor shows the code for '1.c', which includes a bubble sort function and a main function. A warning message at the top states: '이 파일은 프로젝트 타겟에 포함되지 않으므로 코드 분석 기능이 제대로 작동하지 않을 수 있습니다.' (This file is not included in the project target, so code analysis functionality may not work properly.)

```
1 void bubblesort(int *arr, int size);
2
3 int main() {
4     int nicearray[6] = { [0]: 10, [1]: 9, [2]: 8, [3]: 7, [4]: 6, [5]: 5};
5
6     bubblesort(arr, nicearray, sizeof(nicearray)/sizeof(int));
7
8     return 0;
9 }
10
11 void bubblesort(int *arr, int size) {
12     int i, j;
13     for (i = 0; i < size - 1; i++) {
14         for (j = i + 1; j < size; j++) {
15             if (arr[i] > arr[j]) {
16                 int tmp = arr[j];
17                 arr[j] = arr[i];
18                 arr[i] = tmp;
19             }
20         }
21     }
22 }
```

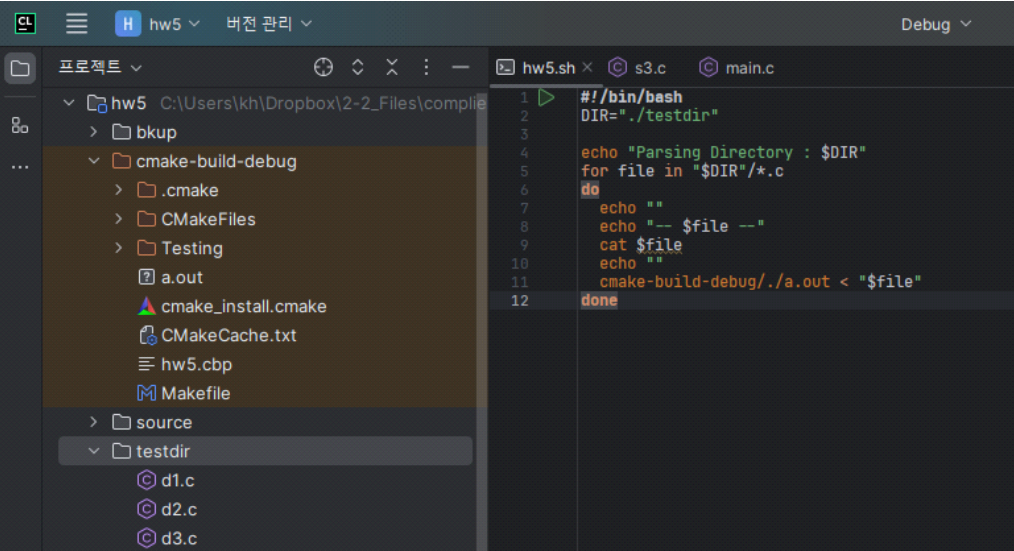
The bottom panel shows the execution output for 'a.out'. It displays a detailed list of symbols and their properties, including types, initialization, and memory addresses. The output is organized into columns for TYPE, INIT, and various symbol names like N\_INIT\_LIST\_ONE, N\_EXP\_ARRAY, N\_EXP\_IDENTITY, N\_STMT\_LIST, N\_STMT\_EXPRESSION, N\_EXP\_ASSIGN, N\_EXP\_ARRAY, N\_EXP\_IDENTITY, N\_STMT\_LIST\_NIL, and N\_STMT\_LIST\_NIL.

종료 코드 0(으)로 완료된 프로세스

hw5 > testdir > 1.c

이제는 정상적으로 작동하는 듯 하여 테스트를 진행하겠다.

4. 테스트



과제 디렉터리 구조는 왼쪽과 같다. 오른쪽은 ./testdir 안의 모든 .c 파일에 대해 일괄적으로 테스트하는 shell script이다. 해당 script로 모든 테스트 파일에 대한 테스트 결과를 redirect한 텍스트 파일(output.txt)은 별개의 파일로 첨부한다.

4-1. 선언(declaration)

d1. 초기화 선언자

```
-- ./testdir/d1.c --
int i1 = 1, i2 = 2;
float f = 3.4;
char c = 'x';
char* nicestring = "this is string";
typedef struct __ns{
    int si;
    int sf;
} nicestruct;
typedef union __nu{
    int ui;
    int uf;
} niceunion;
enum e{
    MON, TUE, WED, THU, FRI, SAT, SUN
} niceenum;
===== syntax tree =====
N_PROGRAM (0,0)
| (ID="i1") TYPE:bb2882f0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
```

```
| | TYPE
| | | (int)
| | INIT
| | | N_INIT_LIST_ONE (0,0)
| | | | N_EXP_INT_CONST (0,0)
| | | | | 1
| (ID="i2") TYPE:bb2882f0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | TYPE
| | | (int)
| | INIT
| | | N_INIT_LIST_ONE (0,0)
| | | | N_EXP_INT_CONST (0,0)
| | | | | 2
| (ID="f") TYPE:bb288380 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | TYPE
| | | (float)
| | INIT
| | | N_INIT_LIST_ONE (0,0)
| | | | N_EXP_FLOAT_CONST (0,0)
| | | | | 3.4
| (ID="c") TYPE:bb288410 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | TYPE
| | | (char 1)
| | INIT
| | | N_INIT_LIST_ONE (0,0)
| | | | N_EXP_CHAR_CONST (0,0)
| | | | | 120
| (ID="nicestring") TYPE:bb28ed80 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | TYPE
| | | POINTER
| | | | ELEMENT_TYPE
| | | | | (char 1)
| | | INIT
| | | | N_INIT_LIST_ONE (0,0)
| | | | | N_EXP_STRING_LITERAL (0,0)
| | | | | nicestring
| (ID="nicestruct") TYPE:bb28ef20 KIND:TYPE SPEC=TYPEDEF LEV=0 VAL=0 ADDR=0
| | TYPE
| | | STRUCT
| | | | FIELD
| | | | | (ID="si") TYPE:bb2882f0 KIND:FIELD SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | (int)
| | | | | (ID="sf") TYPE:bb2882f0 KIND:FIELD SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | (int)
| (ID="niceunion") TYPE:bb28f140 KIND:TYPE SPEC=TYPEDEF LEV=0 VAL=0 ADDR=0
| | TYPE
| | | UNION
| | | | FIELD
| | | | | (ID="ui") TYPE:bb2882f0 KIND:FIELD SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | (int)
| | | | | (ID="uf") TYPE:bb2882f0 KIND:FIELD SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | (int)
```

```
| (ID="niceenum") TYPE:bb28f360 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | TYPE
| | | ENUM
| | | ENUMERATORS
| | | | (ID="MON") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | (ID="TUE") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | (ID="WED") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | (ID="THU") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | (ID="FRI") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | (ID="SAT") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | (ID="SUN") TYPE:0 KIND:ENUM_LITERAL SPEC=NULL LEV=0 VAL=0 ADDR=0
```

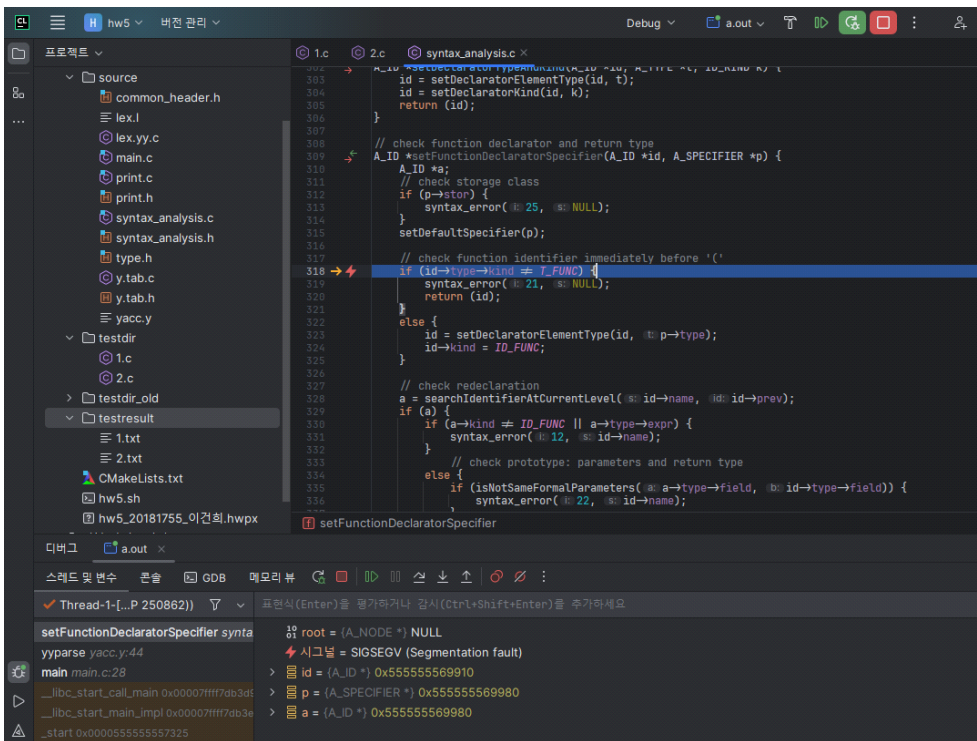
사용 가능한 모든 자료형의 init declarator, typedef의 새로운 자료형 정의에 대해 정상적으로 트리를 출력한다.

## d2. 정의되지 않은 자료형 사용

longint li:

```
kh@ThinkPad-T16g2:~/compiler/hw5/cmake-build-debug$ ./a.out < ../testdir/d2.c
Segmentation fault (core dumped)
kh@ThinkPad-T16g2:~/compiler/hw5/cmake-build-debug$
```

segfault가 발생해서, ide에서 디버그를 진행하였다.



syntax\_analysis.c의 setFunctionDeclaratorSpecifier()에서 사진과 같이 segfault가 발생했다.

```
30 root = (A_NODE *) NULL
31 시그널 = SIGSEGV (Segmentation fault)
32 id = (A_ID *) 0x555555569910
33 name = (char *) 0x5555555698f0 "longint"
34 kind = (ID_KIND) ID_NULL
35 specifier = (S_KIND) S_NULL
36 level = (int) 0
37 address = (int) 0
38 value = (int) 0
39 init = (A_NODE *) NULL
40 type = (A_TYPE *) NULL
41 line = (int) 1
```

해당 줄은 매개변수로 받은 id->type->kind를 검사하는데 정의되지 않은 자료형을 사용했을 경우 id->type이 NULL이다. 이렇게 된다면 id->type->kind는 당연히 접근할 수 없고, segfault가 발생한다. 이를 방지하려면 id->type가 NULL인지 검사하는 코드가 필요하다.

```
638 case 31:
639     printf("format: 'undefined type for identifier %s', s);
640     break;
```

syntax\_error()에서 정의되지 않은 자료형의 사용은 case 31이다. 어떤 정의되지 않은 자료형은 이전과 같이 잘못 파싱되었을 경우 id->name에 저장된다.

```
308 // check function declarator and return type
309 A_ID *setFunctionDeclaratorSpecifier(A_ID *id, A_SPECIFIER *p) {
310     A_ID *a;
311     // check storage class
312     if (p->stor) {
313         syntax_error(25, s: NULL);
314     }
315     setDefaultSpecifier(p);
316
317     if (id->type == NULL){
318         syntax_error(31, s: id->name);
319     }
320 }
```

따라서 setFunctionDeclaratorSpecifier()의 사진과 같은 부분에

1. id->type이 NULL인지 검사하고
  2. NULL이라면 syntax\_error(31, 정의되지 않은 자료형)
- 을 하는 코드를 추가했다.

```
-- ./testdir/d2.c --
```

longint li:

```
** SYNTAX_ERROR ** line 1: syntax error: undefined type for identifier longint near li
```

정의되지 않은 자료형이라 정상적으로 오류를 출력한다. (case 31)



d3. 어떤 identifier를 재정의

```
-- ./testdir/d3.c --
int main(){
    int x = 1;

    char x = 'c';
}
** SYNTAX_ERROR ** line 4: syntax error: redeclaration of identifier x near ;
```

x에 대해 재정의 하였다 오류를 출력한다. (case 12)

d4. 정의되지 않은 identifier 사용

```
-- ./testdir/d4.c --
int main(){
    int ret;

    ret = 1 + x;
}
** SYNTAX_ERROR ** line 4: syntax error: undefined identifier x near x
```

정의되지 않은 identifier x를 사용하였다 오류를 출력한다. (case 13)

d5. 완전하지 않은 forward reference

```
-- ./testdir/d5.c --
struct s;

int main(){
    struct s nicestruct;
}
** SYNTAX_ERROR ** line 5: syntax error: incomplete forward reference for identifier s at end
```

checkForwardReference()는 코드의 끝에서도 실행된다(c언어의 전체 코드는 사실 중괄호로 묶여있음, checkForwardReference()는 매 닫는 중괄호에서 실행됨). struct s에 대한 정보를 전방참조하려 했으나 정의 되어있는 것이 없으므로 이에 대한 오류를 출력한다. (case 32)

4-2. 수식(expression)

e1. 후위 수식 - 배열, 함수 호출, 구조체 멤버 접근, 후위 증가/감소

```
-- testdir/e1.c --
struct pos {
    int xpos;
    int ypos;
};

void fun(int t) {
    int i;
    for (i = 0; i < t; i++) {
        printf("function called! %d\n", i + 1);
    }
}

int main() {
    struct pos p = {37, 80};
    struct pos *pp = &p;
    int a, b;

    int int_array[4] = {
        1, 2, 4, 8
    }; // 배열

    fun(3); // 함수호출

    a = p.xpos; // 구조체멤버 접근 (period)
    b = pp->ypos; // 구조체멤버 접근 (arrow)

    a++; // 후위증가
    b--; // 후위감소

    return 0;
}

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="fun") TYPE:3bbe7bd0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | | (ID="t") TYPE:3bbe12f0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | | (int)
| | | | | TYPE
| | | | | (void)
| | | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | | (ID="t") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | (int)
| | | | | | N_STMT_LIST (0,0)
| | | | | | N_STMT_FOR (0,0)
```

```
| | | | | N_FOR_EXP (0,0)
| | | | | N_EXP_ASSIGN (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="t") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_INT_CONST (0,0)
| | | | | | 0
| | | | | N_EXP_LSS (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="t") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="t") TYPE:3bbe12f0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_POST_INC (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="t") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_STMT_COMPOUND (0,0)
| | | | | N_STMT_LIST (0,0)
| | | | | N_STMT_EXPRESSION (0,0)
| | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="printf") TYPE:3bbe15c0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | N_ARG_LIST (0,0)
| | | | | N_EXP_STRING_LITERAL (0,0)
| | | | | | printf
| | | | | N_ARG_LIST (0,0)
| | | | | N_EXP_ADD (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="t") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_INT_CONST (0,0)
| | | | | | 1
| | | | | N_ARG_LIST_NIL (0,0)
| | | | | N_STMT_LIST_NIL (0,0)
| | | | | N_STMT_LIST_NIL (0,0)
| (ID="main") TYPE:3bbe83d0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | TYPE
| | | | | (int)
| | | | BODY
| | | | N_STMT_COMPOUND (0,0)
| | | | | (ID="p") TYPE:3bbe6960 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | | STRUCT
| | | | | | | FIELD
| | | | | | | (ID="xpos") TYPE:3bbe12f0 KIND:FIELD SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | | (int)
| | | | | | | (ID="ypos") TYPE:3bbe12f0 KIND:FIELD SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | | (int)
| | | | | | INIT
| | | | | N_INIT_LIST (0,0)
| | | | | N_INIT_LIST_ONE (0,0)
| | | | | N_EXP_INT_CONST (0,0)
| | | | | | 37
| | | | | N_INIT_LIST (0,0)
```

```
| | | | | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 80
| | | | | | | | | | N_INIT_LIST_NIL (0,0)
| | | | | | (ID="pp") TYPE:3bbe86a0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | POINTER
| | | | | | | ELEMENT_TYPE
| | | | | | | (DONE:3bbe6960)
| | | | | | | INIT
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_AMP (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="p") TYPE:3bbe6960 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | (ID="a") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | (int)
| | | | | | (ID="b") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | (int)
| | | | | | (ID="int_array") TYPE:3bbe89c0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | ARRAY
| | | | | | | INDEX
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 4
| | | | | | | ELEMENT_TYPE
| | | | | | | (int)
| | | | | | | INIT
| | | | | | | N_INIT_LIST (0,0)
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 1
| | | | | | | N_INIT_LIST (0,0)
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 2
| | | | | | | N_INIT_LIST (0,0)
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 4
| | | | | | | N_INIT_LIST (0,0)
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 8
| | | | | | | N_INIT_LIST_NIL (0,0)
| | | | | | N_STMT_LIST (0,0)
| | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | N_EXP_IDENT (0,0)
| | | | | | | (ID="fun") TYPE:3bbe7bd0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | N_ARG_LIST (0,0)
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 3
| | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | N_STMT_LIST (0,0)
```

```
| | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="a") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | N_EXP_STRUCT (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="p") TYPE:3bbe6960 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | xpos
| | | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="b") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | N_EXP_ARROW (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="pp") TYPE:3bbe86a0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | ypos
| | | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | N_EXP_POST_INC (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="a") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | N_EXP_POST_DEC (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | (ID="b") TYPE:3bbe12f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_RETURN (0,0)
| | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 0
| | | | | | | N_STMT_LIST_NIL (0,0)
```

후위 수식들에 대해 정상적으로 트리를 출력한다.

e2. 전위 수식 - 단항, 캐스팅, 산술, 관계, 논리

```
-- ./testdir/e2.c --
int main() {
    int i1 = 7, i2 = 3;

    // 단항
    int a = i1;
    int b = !i1;
    int c = ++i2;
    int d = --i2;
    int *ptr = &a;
    int e = *ptr;

    // 캐스팅
    float f = (float)i2;
    char g = ((char)i1)+60; // ASCII 67 = 'C'

    // 곱셈 관련
    int h = i1 * i2;
    int i = i1 / i2;
    int j = i1 % i2;

    // 덧셈 관련
    int k = i1 + i2;
    int l = i1 - i2;

    // 관계
    int m = (a < b);
    int n = (a > b);
    int o = (a <= b);
    int p = (a >= b);

    // 논리
    int q = (i1 == i2);
    int r = (i1 != i2);

    return 0;
}

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:70400980 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | TYPE
| | | | (int)
| | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | (ID="i1") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | INIT
| | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 7
| | | | | | | 7
```

```
| | | | | (ID="i2") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | INIT
| | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | 3
| | | | | (ID="a") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | INIT
| | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="i1") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | (ID="b") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | INIT
| | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | N_EXP_NOT (0,0)
| | | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="i1") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | (ID="c") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | INIT
| | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | N_EXP_PRE_INC (0,0)
| | | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="i2") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | (ID="d") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | INIT
| | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | N_EXP_PRE_DEC (0,0)
| | | | | | N_EXP_IDENT (0,0)
| | | | | | (ID="i2") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | (ID="ptr") TYPE:70402110 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | POINTER
| | | | | | | ELEMENT_TYPE
| | | | | | | (int)
| | | | | | | INIT
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_AMP (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | (ID="a") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | (ID="e") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | (int)
| | | | | | | INIT
| | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | N_EXP_STAR (0,0)
| | | | | | | N_EXP_IDENT (0,0)
| | | | | | | (ID="ptr") TYPE:70402110 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
```



```
| | | | | | | | TYPE
| | | | | | | | (int)
| | | | | | | | INIT
| | | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="i1") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="i2") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | (ID="r") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | TYPE
| | | | | | | | (int)
| | | | | | | | INIT
| | | | | | | | N_INIT_LIST_ONE (0,0)
| | | | | | | | N_EXP_NEQ (0,0)
| | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="i1") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="i2") TYPE:703fb2f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | 0
| | | | | | | | N_STMT_LIST_NIL (0,0)
```

전위 수식들에 대해 정상적으로 트리를 출력한다.

4-3. 함수선언문(function\_declaration)

f1. 올바른 함수선언문

```
-- testdir/f1.c --
void printIntFloatChar(int i, float f, char c){
    printf("%d, %f, %c", i, f, c);
}

int main(){
    printIntFloatChar(10, 2.3, 'A');

    return 0;
}

=====  syntax tree  =====
N_PROGRAM (0,0)
| (ID="printIntFloatChar") TYPE:4d0eb30 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | | (ID="i") TYPE:4d092f0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | | (int)
| | | | | (ID="f") TYPE:4d09380 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | | (float)
| | | | | (ID="c") TYPE:4d09410 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | | (char 1)
| | | | TYPE
| | | | | (void)
| | | BODY
| | | | N_STMT_COMPOUND (0,0)
| | | | | N_STMT_LIST (0,0)
| | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="printf") TYPE:4d095c0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_STRING_LITERAL (0,0)
| | | | | | | | | | printf
| | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="i") TYPE:4d092f0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | (ID="f") TYPE:4d09380 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | (ID="c") TYPE:4d09410 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| (ID="main") TYPE:4d0ff0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
```

```
| | | | PARAMETER
| | | | TYPE
| | | | | (int)
| | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="printIntFloatChar") TYPE:4d0eb30 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 10
| | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_FLOAT_CONST (0,0)
| | | | | | | | | | 2.3
| | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_CHAR_CONST (0,0)
| | | | | | | | | | 65
| | | | | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | | N_STMT_LIST (0,0)
| | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | 0
| | | | | | | N_STMT_LIST_NIL (0,0)
```

함수의 모든 파라메터에 대해, 함수에 대해, 함수 호출 시 argument list에 대해, 정상적으로 트리를 출력한다.

f2. 파라메터가 프로토타입과 일치하지 않는 함수의 정의

```
-- testdir/f2.c --
void printNumber(int i);

int main(){
    printNumber(10);

    return 0;
}

void printNumber(char c){
    printf("%c", c);
}

** SYNTAX_ERROR ** line 9: syntax error: conflicting parm type in prototype function printNumber near {
```

파라메터가 프로토타입과 일치하지 않는다 오류를 출력한다. (case 22)

f3. 반환형이 프로토타입과 일치하지 않는 함수의 정의

```
-- testdir/f3.c --
void printNumber(int i);

int main(){
    printNumber(10);

    return 0;
}

int printNumber(int i){
    printf("%d", i);

    return 100;
}

** SYNTAX_ERROR ** line 9: syntax error: illegal or conflicting return type in function printNumber near {
```

반환값의 자료형이 일치하지 않는다 오류를 출력한다. (case 26)

f4. 이름이 없는 파라메터

```
-- testdir/f4.c --
int retint(char){
    return 7.7;
}

int main(){
    int i = retint('x');

    return 0;
}

** SYNTAX_ERROR ** line 1: syntax error: empty parameter name near {
```

파라메터의 이름이 없다 오류를 출력한다. (case 23)



4-4. 명령문 (statement)

s1. 복합문

```
-- testdir/s1.c --
int main(){
    int x;
    x = 0;

    {
        x = x + 2;
        x = x * 2;
    }
}

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:ab8fe980 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | TYPE
| | | | (int)
| | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | (ID="x") TYPE:ab8f92f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | (int)
| | | | | N_STMT_LIST (0,0)
| | | | | N_STMT_EXPRESSION (0,0)
| | | | | N_EXP_ASSIGN (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | (ID="x") TYPE:ab8f92f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_INT_CONST (0,0)
| | | | | 0
| | | | | N_STMT_LIST (0,0)
| | | | | N_STMT_COMPOUND (0,0)
| | | | | N_STMT_LIST (0,0)
| | | | | N_STMT_EXPRESSION (0,0)
| | | | | N_EXP_ASSIGN (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | (ID="x") TYPE:ab8f92f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_ADD (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | (ID="x") TYPE:ab8f92f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_INT_CONST (0,0)
| | | | | 2
| | | | | N_STMT_LIST (0,0)
| | | | | N_STMT_EXPRESSION (0,0)
| | | | | N_EXP_ASSIGN (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | (ID="x") TYPE:ab8f92f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_MUL (0,0)
| | | | | N_EXP_IDENT (0,0)
| | | | | (ID="x") TYPE:ab8f92f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | N_EXP_INT_CONST (0,0)
```

```
| | | | | | | | | | | | | | 2
| | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | N_STMT_LIST_NIL (0,0)
```

main()내의 compound\_statement에 대해 정상적으로 트리를 출력한다.

s2. 선택문(switch) + 레이블된 명령문 + 분기문

```
-- testdir/s2.c --
int sw(int select){
    int sel = select % 3;

    switch(sel) {
        case 0:
            return 100;
            break;
        case 1:
            return 101;
            break;
        case 2:
            return 102;
            break;
        default:
            return 0;
    }
}

int main(){
    int a, b;
    a = sw(1248);
    b = sw(8475);

    printf("%d, %d", a, b);
}

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="sw") TYPE:fd45ca10 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | (ID="select") TYPE:fd4572f0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | (int)
| | | | | TYPE
| | | | | (int)
| | | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | (ID="sel") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | (int)
| | | | | INIT
| | | | | N_INIT_LIST_ONE (0,0)
| | | | | N_EXP_MOD (0,0)
```

```
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | (ID="select") TYPE:fd4572f0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 3
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_SWITCH (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | (ID="sel") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_LABEL_CASE (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 0
| | | | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 100
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_BREAK (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_LABEL_CASE (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 1
| | | | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 101
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_BREAK (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_LABEL_CASE (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 2
| | | | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 102
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_BREAK (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_LABEL_DEFAULT (0,0)
| | | | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | 0
| | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | N_STMT_LIST_NIL (0,0)
| (ID="main") TYPE:fd45e480 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | TYPE
| | | | | (int)
| | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | | (ID="a") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | | (int)
| | | | | | (ID="b") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | TYPE
```

```
| | | | | | | | | | (int)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="a") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="sw") TYPE:fd45ca10 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | 1248
| | | | | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="b") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="sw") TYPE:fd45ca10 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | 8475
| | | | | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="printf") TYPE:fd4575c0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_STRING_LITERAL (0,0)
| | | | | | | | | | | printf
| | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="a") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="b") TYPE:fd4572f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | | | | | N_STMT_LIST_NIL (0,0)
```

switch문에 대한 N\_STMT\_SWITCH, 각 case의 labeled statement인 N\_STMT\_LABEL\_CASE, return에 대한 N\_STMT\_RETURN, break에 대한 N\_STMT\_BREAK등을 포함하여 정상적으로 트리를 출력한다.

s3. 선택문(if~else) + 반복문

```
-- testdir/s3.c --
int fibo(int n) {
    if (n < 2) {
        if (n >= 0) { // n is 0 or 1
            return n;
        }
        else { // n is neg
            return -1;
        }
    }
    else {
        int f0, f1, ans, i;
        f0 = 0;
        f1 = 1;

        for (i = 2; i <= n; i++) {
            ans = f0 + f1;
            if (i == n) {
                return ans;
            }

            f0 = f1;
            f1 = ans;
        }
    }
}

int main() {
    int fibonacci_array[10];

    int i;
    i = 0;

    while (i < 10) {
        fibonacci_array[i] = fibo(i);
        i++;
    }

    i = 0;
    do {
        printf("fibo %d : %d\n", i, fibonacci_array[i]);
        i++;
    } while (i < 10);
}

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="fibo") TYPE:a168ea10 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | | (ID="n") TYPE:a16892f0 KIND:PARAM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | | (int)
| | | | | TYPE
```

```
| | | | | (int)
| | | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_IF_ELSE (0,0)
| | | | | | | | N_EXP_LSS (0,0)
| | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | (ID="n") TYPE:a16892f0 KIND:PARAM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | 2
| | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | N_STMT_IF_ELSE (0,0)
| | | | | | | | | | N_EXP_GEQ (0,0)
| | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="n") TYPE:a16892f0 KIND:PARAM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | 0
| | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | (ID="n") TYPE:a16892f0 KIND:PARAM SPEC=NULL LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | N_STMT_RETURN (0,0)
| | | | | | | | | | | | | N_EXP_MINUS (0,0)
| | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | 1
| | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | | (ID="f0") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | TYPE
| | | | | | | | | | | | (int)
| | | | | | | | | | | (ID="f1") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | TYPE
| | | | | | | | | | | | (int)
| | | | | | | | | | | (ID="ans") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | TYPE
| | | | | | | | | | | | (int)
| | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | TYPE
| | | | | | | | | | | | (int)
| | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | (ID="f0") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | 0
| | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | N_EXP_IDENT (0,0)
```



```
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | 0
| | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | N_STMT_DO (0,0)
| | | | | | | | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="printf") TYPE:a16895c0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | | | | | | N_EXP_STRING_LITERAL (0,0)
| | | | | | | | | | | | | | | | printf
| | | | | | | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | | | | | | N_EXP_ARRAY (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="fibonacci_array") TYPE:a1691120 KIND:VAR SPEC=AUTO LEV=1
VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_ARG_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | N_EXP_POST_INC (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | N_EXP_LSS (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i") TYPE:a16892f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | 10
| | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
```

if~else문의 N\_STMT\_IF, N\_STMT\_IF\_ELSE, iteration\_statement의 N\_STMT\_FOR, N\_STMT\_WHILE, N\_STMT\_DO 등을 포함하여 정  
상적으로 트리를 출력한다.

4-5. 기타

t1. current\_level 테스트

```
-- testdir/t1.c --
int i0;

int main(){
    int i1;
    {
        int i2;
        {
            int i3;
            {
                int i4;
                {
                    int i5;
                    {
                        i0 = 5;
                        i1 = 4;
                        i2 = 3;
                        i3 = 2;
                        i4 = 1;
                        i5 = 0;
                    }
                }
            }
        }
    }
}

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="i0") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | TYPE
| | | (int)
| (ID="main") TYPE:b26aba20 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | TYPE
| | | | (int)
| | | | BODY
| | | | N_STMT_COMPOUND (0,0)
| | | | | (ID="i1") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | TYPE
| | | | | (int)
| | | | | N_STMT_LIST (0,0)
| | | | | N_STMT_COMPOUND (0,0)
| | | | | | (ID="i2") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | TYPE
| | | | | | (int)
| | | | | | N_STMT_LIST (0,0)
| | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | (ID="i3") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=3 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | (int)
| | | | | | | N_STMT_LIST (0,0)
| | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | (ID="i4") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=4 VAL=0 ADDR=0
| | | | | | | | TYPE
| | | | | | | | (int)
| | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | (ID="i5") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=5 VAL=0 ADDR=0
| | | | | | | | | TYPE
| | | | | | | | | (int)
| | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="i0") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | 5
| | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | (ID="i1") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | 4
| | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | (ID="i2") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | 3
| | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | (ID="i3") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=3 VAL=0 ADDR=0
| | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | 2
| | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | (ID="i4") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=4 VAL=0 ADDR=0
| | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | 1
| | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | (ID="i5") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=5 VAL=0 ADDR=0
| | | | | | | | | | | | | | | TYPE
```

```
| | | | | | | | | | | | (int)
| | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | | | (ID="i4") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=4 VAL=0 ADDR=0
| | | | | | | | | | | | TYPE
| | | | | | | | | | | | (int)
| | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | | | | (ID="i5") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=5 VAL=0 ADDR=0
| | | | | | | | | | | | | TYPE
| | | | | | | | | | | | | (int)
| | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | N_STMT_COMPOUND (0,0)
| | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | (ID="i0") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
| | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | 5
| | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | (ID="i1") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | | 4
| | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | | (ID="i2") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=2 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | | | 3
| | | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | | | (ID="i3") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=3 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | | | | 2
| | | | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | | | | (ID="i4") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=4 VAL=0 ADDR=0
| | | | | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | | | | | 1
| | | | | | | | | | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | | | | | | | | (ID="i5") TYPE:b26a52f0 KIND:VAR SPEC=AUTO LEV=5 VAL=0 ADDR=0
```

```
ADDR=0
| | | | | | | | | | | | | | | | | | | | N_EXP_INT_CONST (0,0)
| | | | | | | | | | | | | | | | | | | | 0
| | | | | | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)
| | | | | | | | | | | | | | | | | | | | N_STMT_LIST_NIL (0,0)

i0~i5에서 current_level이 오르기에, LEV도 올라가는 것을 볼 수 있다.
```

```
t2. 스코프 테스트

-- testdir/t2.c --
int i0;

int main() {
    int i1;
    {
        int i2;
        {
            i0 = 10;
            i1 = 11;
            i2 = 12;

        }

        i2 = i2 + 100; // 위치 1
    }

    i0 = i0 + 100;
    i1 = i1 + 100;
    i2 = i2 + 100; // 위치 2
}

** SYNTAX_ERROR ** line 18: syntax error: undefined identifier i2 near i2

위치 1은 i2의 스코프 내이기에 접근이 가능하지만, 위치 2는 i2의 스코프 밖이기에 접근이 불가능하여 정의되지 않은 이름이라 오류를 출력한다.
```